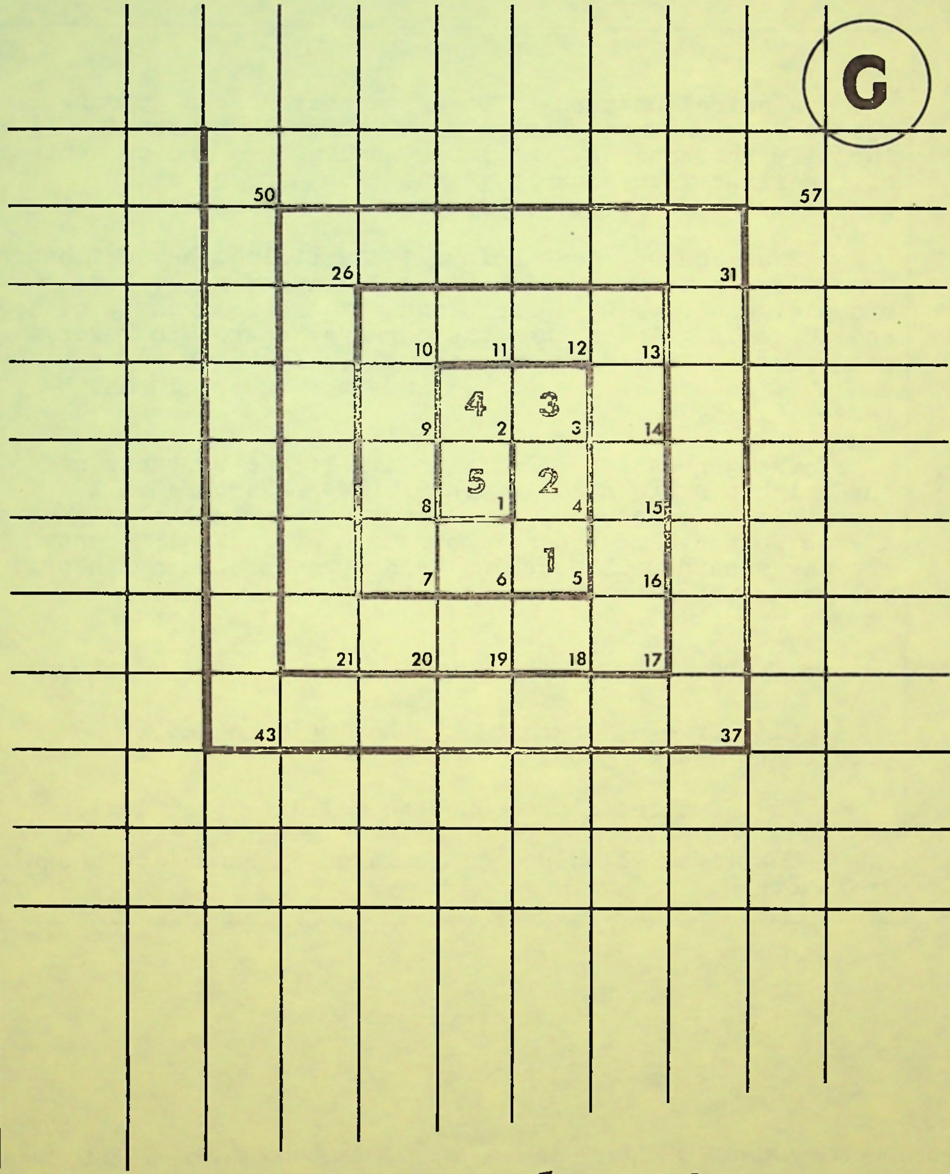
9

Popular Computing

BOX 272 CALABASAS, CA 91302

VOL 1 NO 9 DEC 73



square spiral

PROBLEM 24

The Square Spiral Problem

A spiral pattern is to be generated on a grid of squares, as shown in Figure G. The small numbers identify the squares; the large numbers are the contents of the first five squares. The five numbers shown constitute the initial conditions of the problem.

Each square, just before it is filled, has neighbors. Generally, a square will have four neighbors, as for example, square #19, whose neighbors are squares 7, 6, 5, and 18. In going around the corners, there are fewer neighbors. Square 16, for example, has three (4, 5, and 15); square 17 has two (5 and 16); square 18 has three (5, 16, and 17).

The generation rule is this: if the contents of the neighbors are all different, the new square will have twice the value of the lowest number in the neighbors. If the contents of the neighbors are not all different, the new square will contain the average of the neighbors' values, rounded to the nearest integer.

The problem is this: What value will go into square #5000?

(1) Draw a flowchart of the logic needed to proceed to square 5000.

(2) Outline a procedure to validate that logic.

The first 91 values (calculated by hand) are shown in Figure H.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $15 per year, or $12 if remittance accompanies the order. For all foreign subscriptions, add $4.50 per year. Multiple subscriptions to the same address, add $5 each; thus, 3 copies per month is $25 per year, U.S. delivery. Back issues $1.50 each. Subscriptions may begin with any issue. Subscriptions for qualified undergraduate students half price. Copyright 1973 by POPULAR COMPUTING.

Publisher: Fred Gruenberger
Editor: Audrey Gruenberger
Associate editor: David Babcock
Contributing editors: Richard Andree, Paul Armer, Daniel D. McCracken, William C. McGee
Technical editor: James J. Johnson
Advertising manager: Ken W. Sims
Art director: John G. Scott

@ 2023 This work is licensed under CC BY-NC-SA 4.0

H

8	8	5	4	4	4	4	5	10	20
5	4	4	4	4	4	4	8	16	
8	4	4	4	4	4	5	10	16	
10	5	4	4	4	4	6	10	12	
16	8	4	4	4	3	6	7	14	
14	10	5	4	5	2	4	8	14	
14	7	6	4	2	1	2	4	8	
9	8	6	3	2	2	2	3	6	
10	8	4	2	2	3	4	4	6	
10	5	4	3	4	6	5	8	8	

The magic numbers for 1974

The following table gives the magic numbers for dates in the year 1974. To find the day of the week, add the day of the month to the number indicated and reduce the result modulo 7. Sunday is 0; Monday is 1; Tuesday is 2;...,Saturday is 6. For example, Christmas day will be on $6 + 25 \equiv 3 \pmod 7$, which is Wednesday.

JAN	1	APR	0	JUL	0	OCT	1
FEB	4	MAY	2	AUG	3	NOV	4
MAR	4	JUN	5	SEP	6	DEC	6

N-Series

9	
ln	2.1972245773362193827904904738450514092949811156454989O
log	.95424250943932487459005580651023061840025772838139173
99	
ln	4.59511985013458992685243405181018070911668796958291608
log	1.99563519459754991534025577775325486010695991884784482
999	
ln	6.90675477864855351855383138179902427784254926063588452
log	2.99956548822598230869353439930444753755833592827384014
9999	
ln	9.21024036697584937773663231872316986600757055678052083
log	3.99995656838019248961544395597619277332624927405429742
99999	
ln	11.5129154649202280867541239200883210365769235144613242
log	4.99999565703346609862064785135359168695879946192655354
999999	
ln	13.8155095579637741037746151447726519121066087889153700
log	5.99999956570530094936245578708246424160336305791588691
9999999	
ln	16.1180955509583147881256068494322161178743769204013965
log	6.99999995657054963820226295378981140583617647934838068
99999999	
ln	18.4206807339523654221439313041415778274754585756966838
log	7.99999999565705515925275748356129104145734723683018995
.9	
ln	-.10536051565782630122750098083931279830612037298327407
log	-.04575749056067512540994419348976938159974227161860827
.99	
ln	-.01005033585350144118354885755854770608551500767462987
log	-.00436480540245008465974422224674513989304008115215518
.999	
ln	-.00100050033358353350014298225406834496075520525043441
log	-.00043451177401769130646560069552462441664071726159986
.9999	
ln	-.000100005000333358335333350001428696439683539773457108
log	-.00004343161980751038455604402380722667375072594570258
.99999	
ln	-.00001000005000033333583335333350000142858392868254068
log	-.00000434296653390137935214864640831304120053807344646
.999999	
ln	-.000001000000500000333333583333533333350000014285726786
log	-.00000043429469905063754421291753575839663694208411309
.9999999	
ln	-.00000010000000500000033333335833333533333350000001429
log	-.00000004342945036179773704621018859416382352065161932

4 Subroutines PROBLEM 25

There are four subroutines available. Each of them outputs a stream of non-decreasing integers, one integer per call of the subroutine. The subroutines are independent of each other.

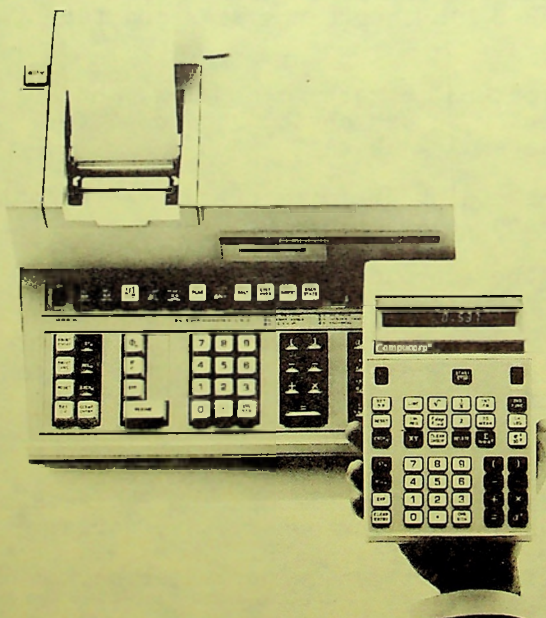
The output of all four subroutines is to be outputted (to tape, say) in ascending order with no repeats; that is, no number is to be duplicated in the output stream.

(A) Draw a flowchart for the logic of this situation.

(B) Devise a procedure to test a debugged program that follows the logic of (A).

The test procedure is for the logic of the merging operation. Thus, for purposes of the test, specific subroutines can be inserted.

Complex international problems solved while you wait.



Compucorp's extensive line of specialized desktop and hand-held computers are providing instant answers to complex questions in over 60 different countries. No matter what language you speak, we understand your problems.

Compucorp.
Computer Design Corporation
12401 W. Olympic Blvd.
Los Angeles, California 90064
(213) 478-9761

Buenos Aires
49-2949

Hong Kong
(H) 792-161

London
01-579-5542

Zug, Switzerland
(042) 21 60 17

14th Symposium

Every year since 1958, a one-day discussion session on computing has been held in the Fall (the first ten such sessions were held at the RAND Corporation). At the 1972 session, the attendees were:

Prof. Richard Andree
University of Oklahoma

Paul Armer
Center for Advanced Study in the Behavioral Sciences

Mort Bernstein
System Development Corp.

Dr. Ned Chapin
InfoSci

Prof. Fred Gruenberger
California State University, Northridge

Dr. Richard Hamming
Bell Laboratories

Robert L. Patrick
Computer Specialist

Prof. Stan Rifkin
California State University, Northridge

Ed Ryan
Union Oil Company

Prof. Robert Teague
California State University, Northridge

Dr. Roger van Norton
University of Arizona and National Science Foundation

Robert White
Informatics Inc.

The chief topic for discussion was "Complexity and Communication," with the following agenda:

1. How should we communicate methods of solution in computing? The literature suggests flowcharts as a good tool of communication, but our own preliminary research for this meeting indicates that flowcharts are weak and far from standardized.

2. The use of computers is getting complex, and perhaps unmanageably complex. Can anything be done to avert what seems to be impending chaos?

3. If the planning and programming of large systems is getting too complex, and we can't communicate very well, then where do we go? How do we teach this crazy business to the next generation, when us oldtimers can't make too much sense of it any longer? Or can it be that the youngsters <u>do</u> understand it, and we don't?

As a starting point, each of the attendees was asked to prepare, in advance of the meeting in Anaheim, a flowchart for the following textbook problem:

Given 960 words in core storage, addressed at T, T+1, T+2,...,T+959. These 960 words make up 240 sets of four words. For each set of four consecutive words, the contents should be of four types, A, B, C, and D. The specific character of these types is not material.

We wish to establish that each of the 240 set of four words contains all four types in the order ABCD. For any set of four words that does not contain all four types, we wish to print out the set number. If a set of four words contains the four types in some order other than ABCD, we wish to put the contents in the correct order.

For example, in the following situations:

B	A	E	C
T+64	T+65	T+66	T+67
B	A	B	C
T+120	T+121	T+122	T+123
D	B	A	C
T+88	T+89	T+90	T+91

the first set should cause the printing of "Set 17," the second should cause the printing of "Set 31," and the third set should be rearranged in storage in the order ABCD.

This exercise seemed to indicate that the use of flowcharts as a tool of communication (which use is widely advocated in textbooks) is a myth. Of the flowcharts produced by the experts, no two used similar notation; few of them could be read by other than their author; and their physical sizes varied by a factor of five. (A possible flowchart is given at the end of this article.)

So the flowcharting exercise was largely just embarrassing (just one attendee validated his flowchart by coding from it and testing that code). It did trigger a lively discussion for the day, though. A copy of the complete transcript can be obtained for $5 from the Bureau of Business Services and Research, California State University, Northridge, 91324, the sponsor of the symposium. Some scattered excerpts follow.

Hamming: The universities aren't turning out a product that the country wants. I hear people in industry say "I will never again hire a computer science major." The universities should not be vocational schools, but on the other hand, they shouldn't turn out completely useless people, either.

Andree: It's been suggested that we train only the bright ones and smother the others, but how do you choose which is which?

van Norton: There are only two kinds of people: the right kind and the wrong kind, and only the right kind know which is which.

Hamming: There's another problem. If you fail in many fields--math, physics, engineering, etc.--you can then go into computer science. We're getting a large collection of second rate people. We need a selection system to get first rate intelligent people, and graduate a low volume of those.

Patrick: The balance sheet [on people] doesn't tell the whole story. True, the total head count goes down. We're getting rid of the type of employee who has high turnover, high error rate, expensive training, and who is troublesome. We're going to a smaller crew of more sophisticated people. The total hardware costs are, in fact, going down. But the resulting environment is extremely high pressure; the ulcer rate, the divorce rate, and the nerve-jangling rate are taking us right to the edge. We can't handle the new environment intellectually. We have crisis after crisis in which everybody works all weekend to recover from a crash situation. The situation is similar to that of the air traffic controllers. We are using up our reserve people. And all this tells me that we shouldn't build one more level on top of all that.

We shouldn't build complex systems until we can structure them properly. But my clients listen to the salesmen and plunge in to get a new system up and running within a year. That year includes all the programming and the equipment installation--and then they are astonished to find it doesn't work properly. It may be that we aren't planning in a way that is commensurate with the size of our ventures.

Hamming: The course I want to teach sometime is called "Great Ideas in Software." This would consist of discussions--not in great depth--of the significant topics in software.

[The entire group came up with the following list of topics for the course Hamming suggested.] Finite vs. infinite precision. Loops and subroutines. Levels of language. Restart and recovery procedures. Debugging and testing of programs. Self-descriptive languages. Self-descriptive files. List processing. Arrays. Parallel vs. sequential processing. Scheduling; monitors; resource allocation. Virtual storage; virtual machines. Reentrant and relocatiable code. Address modification. Time-sharing. Floating point. Push down lists.

Hamming: The average faculty member in computer science couldn't pass the courses that the undergraduates are taking. Of course, that's also true in other disciplines. My personal conviction is that a solid grounding in statistical theory is of the essence.

Weiss: Let me reduce it to three things that should be taught. The first is ability to communicate; we're all agreed on that. The second is how to compute, and I care not what language or languages you use. And the third is abstract thinking; what has been called problem solving, which includes planning, estimating, and broad-brushing. It's the side of engineering that has economics in it, but the economics must be disguised.

Patrick: Some years back, IBM paid me to write a history of a software project, which may have been a first in the field and may still be the only one. This is in sharp contrast to engineering work, where project histories are considered normal. When a new engineering project comes along, the project manager can usually find a

project history of a similar project. I don't know of any group in software that comes even close to being that well organized.

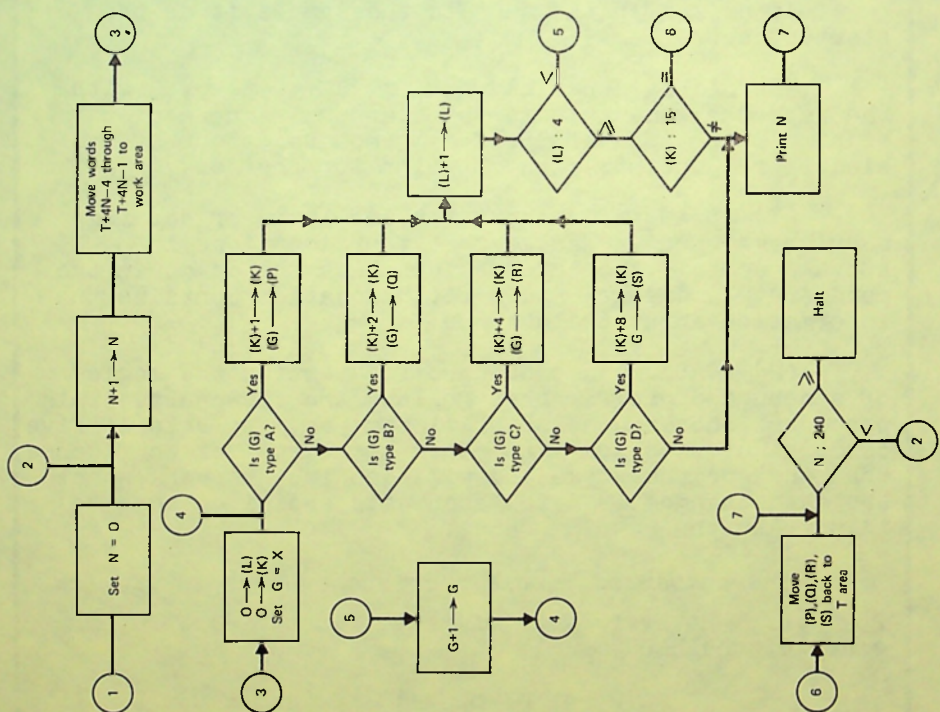
White: Mark Twain's remark seems most appropriate at this point: "First God made idiots. That was for practice. Then He made school boards."

Patrick: We need to have someone collect in a book all the fundamental ideas, so that people wouldn't have to go back to RAND's LP codes to find out about restart procedures.

Hamming: If someone did, publishers wouldn't publish it, and if they did, the book wouldn't get circulated, and if it did, people wouldn't read it, and if they did, they'd ignore the advice.

Patrick: Well, I'm not sure you should be teaching those things to 22 year old kids. Teach them the basics like how to communicate, and how to devise and implement algorithms, and how to test their programs--that may be enough. These advanced topics should be taught to the 30 year old practitioners in the field, who have evidently never heard of them.

Ryan: Students look at us as though we were geniuses, when we can spot the blunder in their code in a few seconds. I tell them that it isn't genius--it's just that I've been there, many times.



X	X+1	X+2	X+3	(Work area)
P	Q	R	S	(Work area)

Cribbage Scoring PROBLEM 26

A cribbage hand consists of four cards (from a regular deck of 52) plus a fifth, called the starter, which the player does not hold. The 10's and all picture cards count 10 for point value.

Any combination of cards that total 15 scores 2 points.

Pairs score 2 points. Three of a kind (which has three pair combinations) scores 6; four of a kind scores 12.

Each combination that makes a run of three or more scores the number of cards in the run.

If the four cards held are of the same suit, score 4; if the starter is also of the same suit, score 5.

If a jack in the hand is the same suit as the starter, score 1.

The highest possible hand is J, 5, 5, 5, 5 with the jack the same suit as the starter. There are eight combinations of 15 (for 16 points); four of a kind (for 12 points); and 1 point for the jack.

(1) Draw a flowchart for the logic of scoring a cribbage hand. Assume that four locations in storage contain the information about the cards in the hand (rank and suit) and a fifth location contains the information about the starter.

(2) Outline a procedure for testing the logic of a debugged program that follows the flowchart. This procedure should consist of listing as many sets of five cards as are necessary to reach some level of confidence that the program works and will continue to work when the data changes. A few such sets are given here to illustrate the scoring rules.

The number of possible cribbage hands is ${}_{52}C_5 =$ 2598960. However, many of these cluster. For example, the hand

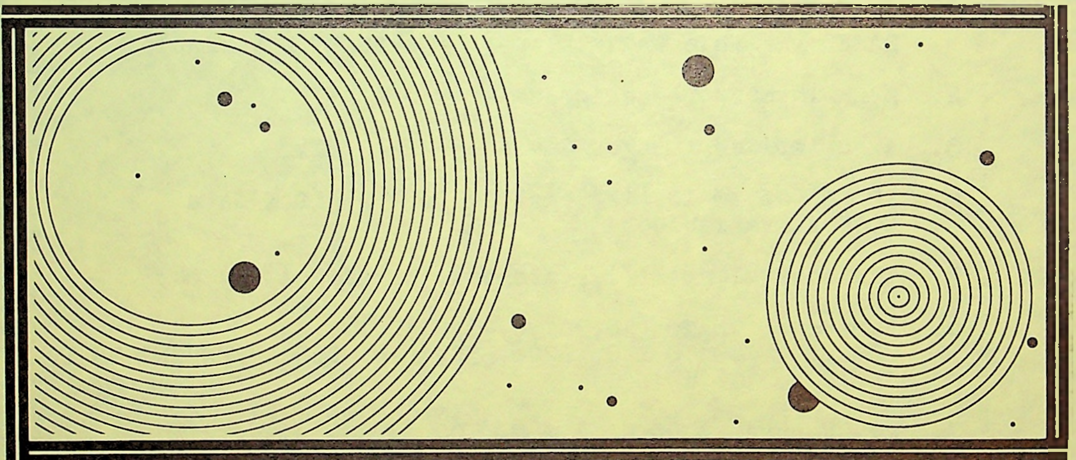
2, 4, 6, 8   10   (score 0)

is the same, for scoring purposes, regardless of the suits involved, so that there are nearly 1024 hands disposed of at once (all except those that make up the

flushes). In any event, it should be relatively easy to run through all possible cribbage hands and produce the probability distribution of the scores.

In hand	Starter	Score
2, 4, 6, J	8	0
2, 4, 6, J*	8*	1
2, 2, 4, 6	8	2
2, 2, 8, 8	A	4
3, 3, 3, J*	4*	7
7, 7, 8, 8	10	12
7, 8, 8, 9	10	14
9, 10, 10, J*	10*	16
6*, 7*, 8*, 9*	7	18
4, 5, 6, 6	6	21
4, 4, 5, 6	6	24
5, 5, 5, 5	10	28

* indicates same suit



In the magazine EDN for September 5, 1973, the article "Pocket Calculator Radiation Can be Heard on Radios and TVs" surveys the noise effects of six machines (TI-2500, TI SR-10, Bowmar, Remington 661, HP-35, and Compucorp 322G) on the broadcast band, FM, shortwave, and TV. In all cases, the calculator must be close to the receiver's antenna.

For most machines, the RF noise comes from the display. The sounds from the HP-80 are similar to those from the HP-35. However, since the HP-80 can be put into a calculating loop (see the review in PC5), and the display is active during this action, the audible effects can be made continuous without hand keying.

Speaking of Languages

ROBERT TEAGUE

In a previous issue (PC6-8) a quiz on programming languages was offered to test coding skills in Fortran, ALGOL, COBOL, and PL/1. Although there were no entries submitted that had all the correct answers, one entry had only one error and was also the first set of answers received. The winner is

Robert Bullock, Jr.
2471 Laclede Station Road
Maplewood, MO 63143

who is applauded for obvious skill in computer languages.

The answers to the ten questions of the quiz are given below.

1. The notation means "K points at A," and A must be a BASED variable while K must be a POINTER variable.

2. ØVER-18 must be assigned level 88.

3. A÷B implies integer division in Algol.

4. IF (K) GØ TØ 10 is legal in Fortran if K is a LOGICAL variable.

5. The equivalent PL/1 statement is: L = (A > B) *4;

6. Fortran: GØ TØ (10,20,30,40,50), I
   CØBØL: GØ TØ L10, L20, L30, L40, L50 DEPENDING ØN I.
   PL/1: GØ TØ A(I);

7. a) Fortran: DØ 5 J = M, 100, 3
   CØBØL: PERFØRM P-1 VARYING J FRØM M BY 3 UNTIL J > 100.
   PL/1: DØ J = M TØ 100 BY 3;

   b) Fortran: DO 5 L = 8, 149, 4
   CØBØL: PERFORM P-1 VARYING L FRØM 8 BY 4 UNTIL L > 149.
   PL/1: DØ L = 8 TØ 150 BY 4;

8. Algol: A:= <u>if</u> B = 4 <u>then</u> 1 <u>else</u> 0;
   CØBØL: IF B = 4 MØVE 1 TØ A ELSE MØVE ZERØ TØ A.
   Fortran: A = MIN(ABS(B-4),1)

9. I = -42 and J = 0 after PL_1 is executed.

10. E(1,2,1) = 03, E(2,1,2) = 06, d(1,1) = 0102, D(2,2) = 0709, and C(2) = 05060708.

This month's quiz consists of the CØBØL program given below. Answers to the three questions listed should be sent to

Speaking of Languages...
POPULAR COMPUTING
Box 272
Calabasas, CA 91302

1) Will the program compile (don't forget to make appropriate changes to the ENVIRONMENT DIVISION)?

2) Will the program execute?

3) How many values are produced? Why?

```
IDENTIFICATIØN DIVISIØN.
PRØGRAM-ID. CØBØL-1.
AUTHØR. R TEAGUE.
ENVIRØNMENT DIVISIØN.
CØNFIGURATIØN SECTIØN.
SØURCE-CØMPUTER. 3300.
ØBJECT-CØMPUTER. 3300.
INPUT-ØUTPUT SECTIØN.
FILE-CØNTRØL.  SELECT PRINT-FILE ASSIGN TØ SYSTEM-ØUTPUT.
DATA DIVISIØN.
FILE SECTIØN.
FD   PRINT-FILE LABEL RECØRDS ARE ØMITTED.
01   PRINT-LINE.
     02 FILLER PICTURE X(5).
     02 RESULT PICTURE Z(5).
WØRKING-STØRAGE SECTIØN.
77   A PICTURE 9(5).
77   B PICTURE 9(5).
77   C PICTURE 9(5).
77   N PICTURE 9(5).
PRØCEDURE DIVISIØN.
BEGIN.
     MØVE 1 TØ A.
     MØVE 1 TO B.
     MØVE 1 TO C.
CØMPUTATIØN.
     PERFØRM CALCULATIØN VARYING N FRØM 1 BY 1 UNTIL N = 100.
CALCULATIØN.
     MØVE B TØ A.
     MØVE C TØ B.
     ADD A,B GIVING C.
     MØVE C TØ RESULT.
     WRITE PRINT-LINE.
HALT.
     CLOSE PRINT-FILE.
     STØP RUN.
```

The Harmonic Series

PROBLEM 27

The series

$$1 + 1/2 + 1/3 + 1/4 + 1/5 + \ldots + 1/k$$

is well known to diverge; indeed, it is the first divergent series presented in most texts. As Martin Gardner points out (in his Sixth book of collected columns from Scientific American), it diverges "with infuriating slowness. The first 100 terms, for instance, total only a bit more than 5. In 1968 John W. Wrench, Jr., calculated the exact number of terms at which the series has a partial sum exceeding 100; the number of terms is

1509 26886 22113 78832 36935 63264 53810 14498 59497."

In the accompanying table, various partial sums for the harmonic series are given. For those marked with an asterisk, the sum has just passed another integral value.

The ratio of the number of terms needed to advance by one in the sum approaches e. Thus, the ratio of the last two values of N in the table is

2.71828 18284 59045 23536 01393

which differs from e by 1 in the 23rd significant digit.

The last value given in the table (for the sum of 50) should thus, when multiplied by e^{50} ($5.184705528 \times 10^{21}$) produce Wrench's number given above. Similarly, the number of terms of the harmonic series needed to yield a partial sum exceeding 200 should be 4.0570916×10^{86}.

In a computer atmosphere, where word lengths are fixed and arithmetic is finite, the harmonic series converges. To what does it converge, in a floating arithmetic system of

A) 6 significant digits?
B) 8 significant digits?
C) 10 significant digits?
D) 15 significant digits?

N		$\sum_{k=1}^{N} (1/k)$
4	*	2.08333 33333
10		2.92896 82539
11	*	3.01987 73448
20		3.59773 96571
30		3.99498 71309
31	*	4.02724 51954
40		4.27854 30389
50		4.49920 53383
83	*	5.00206 82726
100		5.18737 75176
200		5.87803 09481
227	*	6.00436 67083
300		6.28266 38802
400		6.56992 96911
500		6.79282 34299
600		6.97497 84219
616	*	7.00127 40971
700		7.12901 01155
1000		7.48547 08605
1674	*	8.00048 55719
2000		8.17836 81036
3000		8.58374 98899
4000		8.87139 02997
4550	*	9.00020 80629
5000		9.09450 88529
10000		9.78760 60360
12367	*	10.00004 30082
20000		10.48072 82172
33617	*	11.00001 77086
91380	*	12.00000 30516
248397	*	13.00000 12294
675214	*	14.00000 13620
1835421	*	15.00000 03782
4989191	*	16.00000 00954
13562027	*	17.00000 00148
36865412	*	18.00000 00037
100210581	*	19.00000 00097
272400600	*	20.00000 00016
10708978215761 67177938	*	49.00000 00000
29110020885268 72100231	*	50.00000 00000

Barbeque PROBLEM 28

The directions that come with the new barbeque specify the following times for cooking steaks:

Thickness of steak	Rare			Medium			Well done		
1"	2	3	1	4	4	4	5	6	7
1 1/2"	4	5	2	7	8	5	9	10	8
2"	7	7	3	8	9	6	10	11	9

minutes on second side

minutes on first side

If the chef is to prepare nine steaks, one of each type listed in the chart, and he wishes to have all of them done at the same time, then he should follow this timetable:

Time		Time	
0	#9 on	13	#4 on; turn #5
2	#8 on	14	Turn #3
4	#6 on	15	Turn #7
6	#5 on	16	#1 on; turn #2
7	#3 on	17	Turn #4
10	#7 on; turn #9	18	Turn #1
11	Turn #8	21	Take all off
12	#2 on; turn #6		

Given the timing chart in storage, draw a flowchart for the logic to output a similar timetable for any given combination of steaks to be cooked.

? FRUSTRATED

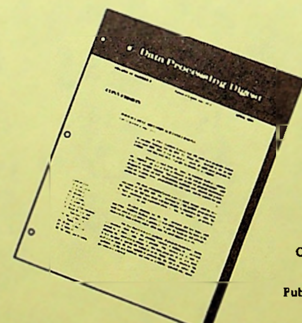
TRYING TO FIND *GOOD* COMPUTER LITERATURE. . .
. . . AND THE TIME TO READ IT?

Hire full-time research for just $4.25 a month! Here's what you get:

1. a staff of computer pros continuously monitoring the computer literature
2. a technical library source of 59 computer publications and 123 trade/management publications
3. news of conferences, meetings, seminars
4. reviews of new books
5. original reports about problems faced and solved, but not yet reported in the literature

. . .presented in report form each month. Write for information about DATA PROCESSING DIGEST. Or send $4.25 for our current issue and apply to your continuing subscription (12 issues, $51).

Data Processing Digest, Inc.
[illegible] LA TIJERA BOULEVARD, LOS ANGELES, CALIFORNIA 90045 / PHONE (213) 776-4334



Abstracts
Digests
Resources
News Items
Calendar
Reviews
Original Reports
Yearly Index
Published Each Month
Since 1955

Name ____________
Dept. ____________
Company ____________
Address ____________
City ________ State ________ Zip ____
PC

PC9-16

	2	3	5	7	11	13	17	19	23	29	31	37	41
0	86	68	58	35	41	14	27	44	10	14	16	16	9
1	91	43	42	30	--	14	13	23	7	3	--	6	--
2	168	59	1	28	38	25	16	22	28	20	22	7	19
3	153	84	23	20	34	7	26	35	26	15	13	14	8
4	107	106	55	29	11	28	23	10	11	26	10	23	13
5	71	91	--	25	28	20	26	27	12	10	19	19	23
6	93	104	45	33	25	15	13	18	11	26	8	10	8
7	71	101	53	39	19	37	10	12	50	45	15	22	19
8	78	100	24	33	27	23	20	21	15	10	28	16	16
9	108	56	65	61	12	8	27	16	9	18	16	25	17

For the base numbers at the top, the body of the table shows the highest power lacking the digit given at the left stub. For example, every power of 29 greater than the 26th contains the digit 4.
Problem: Extend the table for the primes from 43 through 97.

PROBLEM 29

2 Tables

	0	1	2	3	4	5	6	7	8	9
1	10	4	1	5	2	8	4	15	3	12
2	53	40	43	25	18	16	46	24	19	33
3	242	42	43	83	44	41	157	24	39	50
4	377	313	314	219	192	41	220	181	180	421
5	1491	485	314	221	315	973	220	317	316	422
6	1492	1841	2354	2270	3396	973	2269	972	971	2187
7	6801							972		
8	14007									

The table shows the first appearance of adjacent alike digits in powers of 2. The first entry at the upper left indicates that the digit 0 first appears in the 10th power of 2; 8 adjacent zeros first occurs in the 14007th power. Similarly, the first appearance of 6 adjacent 4's occurs in the 3396th power.